	Type	L #	Hits	Search Text	DBs	Time Stamp
1	BRS	L1	1	6195629.pn.	USPAT	2004/04/16 14:26
2	BRS	L2	0	6195629.pn. and override	USPAT	2004/04/16 14:28
3	BRS	L3	1	6560757.pn. and override	USPAT	2004/04/16 14:28
4	BRS	L4	0	6560757.pn. and (override near simulation)	USPAT	2004/04/16 14:28
5	BRS	L5	1	6560757.pn. and (override and simulation)	USPAT	2004/04/16 14:30
6	BRS	L6	0	6195627.pn. and (override and simulation)	USPAT	2004/04/16 14:30
7	BRS	L7	1	6053947.pn. and (override and simulation)	USPAT	2004/04/16 14:32
8	BRS	L8	1	6053947.pn. and override	USPAT	2004/04/16 14:32
9	BRS	L9	1	5095454.pn. and override	USPAT	2004/04/16 14:33
10	BRS	L10	0	5095454.pn. and override and latches and enable and disable	USPAT	2004/04/16 14:34
11	BRS	L11	1	5095454.pn. and override and latches	USPAT	2004/04/16 14:36
12	BRS	L12	0	5943490.pn. and override and latches	USPAT	2004/04/16 14:36
13	BRS	L13	0	5680332.pn. and override and latches	USPAT	2004/04/16 14:37
14	BRS	L14	0	5680332.pn. and override	USPAT	2004/04/16 14:37
15	BRS	L15	0	5103450.pn. and override	USPAT	2004/04/16 14:37
16	BRS	L16	0	5883809.pn. and override	USPAT	2004/04/16 14:37
17	BRS	L17	0	5910897.pn. and override	USPAT	2004/04/16 14:38
18	BRS	L18	0	6182206.pn. and override	USPAT	2004/04/16 14:38
19	BRS	L19	0	6202042.pn. and override	USPAT	2004/04/16 14:39
20	BRS	L20	0	6052524.pn. and override	USPAT	2004/04/16 14:39
21	BRS	L21	0	5920490.pn. and override	USPAT	2004/04/16 14:39
22	BRS	L22	0	5841967.pn. and override	USPAT	2004/04/16 14:40

	Type	L #	Hits	Search Text	DBs	Time Stamp
23	BRS	L23	0	6212491.pn. and override	USPAT	2004/04/16 14:40
24	BRS	L24	1	5544067.pn. and override	USPAT	2004/04/16 14:41
25	BRS	L25	0	5544067.pn. and override and latches	USPAT	2004/04/16 14:41
26	BRS	L26	0	6223142.pn. and override	USPAT	2004/04/16 14:42
27	BRS	L27	0	5812416.pn. and override	USPAT	2004/04/16 14:42
28	BRS	L28	0	5604895.pn. and override	USPAT	2004/04/16 14:42

US-PAT-NO: 5095454
DOCUMENT-IDENTIFIER: US 5095454 A
TITLE: Method and apparatus for verifying timing during simulation of digital circuits

---

Abstract Text - ABTX (1):

A digital circuit simulation method and apparatus provide for critical path timing analysis of digital circuitry using a hybrid path tracing method. The hybrid path tracing performs path tracing when, for example, simulation values change at designated inputs. The path tracing can employ the simulation values for eliminating blocked paths. During tracing, the method and apparatus determine the shortest and longest paths from each input or beginning point to each end point. The end points are typically storage elements such as latches, flip-flops or systems outputs at a high functional level. The critical path tracing analysis finds the shortest and longest paths from the beginning point to the end point and records and saves the violation history, if any, associated with those paths. A timing template allows the user to develop the necessary input stimuli, in a logical ordered format to test the timing behavior of the digital circuit to be designed. Special procedures are also available for enabling the iteration of stimuli to check the timing of a transparent latch-based design. The system thus provides as an output either a summary or detailed history of violations at selected levels within the circuit to be analyzed. User input enables the system to isolate upon particular portions of the circuitry for determining correct operation.

US Patent No. - PN (1):

5095454

Brief Summary Text - BSTX (10):

It is therefore a primary object of the invention to provide a timing analysis method and apparatus which performs substantially complete timing analysis using all reasonable simulation pattern inputs without requiring excessive computer time, while maintaining the number of false violations to a minimum or none. Other objects of the invention are a method and apparatus which provide, using a few timing patterns, two paths through the circuitry which are the minimum and maximum path times from input node to output node Another object of the invention is for enabling a timing analysis to handle logical circuit loops and logical storage elements such as latches or flip-flops.

Brief Summary Text - BSTX (14):

The method of the invention further features allowing the user to designate input stimuli in a timing template for investigating the timing behavior of the digital circuit being designed and providing special latch handling methods for allowing the iteration of stimuli to be used to check the timing of a transparent latch based loop design.

Brief Summary Text - BSTX (17):

In yet another aspect of the apparatus of the invention, circuitry for inputting a user specified timing template is provided and further elements for setting the input stimuli with regard to at least one of the designated inputs, in accordance with the user specified timing template, is employed for determining the timing behavior of the digital circuit being tested. Circuitry is provided also, in a preferred aspect of the invention, for iteration of stimuli at designated inputs for checking the timing of the digital circuit design for a loop containing transparent latches.

Drawing Description Text - DRTX (10):

FIG. 13 illustrates a latch based design in accordance with the invention; and

Detailed Description Text - DETX (3):

The timing behavior of the elements is specified in a timing construct designated a "specify block." Typically, the timing behavior for many elements are well known and are common to a number of digital circuits. Thus, the delays inherent in passing through an AND gate or an OR gate, or passing through a transistor are well known and are among the elements specified in the net list 14. Other elements, however, such as flip-flops, latches, and RAMs, must be specified using the specify blocks. The construction of the specify block is described hereinafter. With the network configuration from elements 12 (including the specify blocks), and the timing templates from elements 16, the method of the invention then employs the circuitry of element 14 to effect the timing analysis. The results of the analysis are the timing violations 18 output from the system for viewing and use by the end user.

Detailed Description Text - DETX (5):

In general operation, once the overall circuit configuration has been input to the system, whether by keyboard, by a previously created file in memory, or otherwise, a simulation program such as Verilog begins to operate. (Verilog is a commercially available simulation analysis system software marketed by Gateway Design Automation Corporation of Lowell, Massachusetts.) After each simulation time or instant of the simulation analysis at the logical or behavioral level, the timing analysis operates using as "landmarks" primary inputs designated by the end user. As signals change at those primary inputs, whether the changes be forced by end user input or occur as a result of a signal change of a simulation value at the input, the timing analysis method propagates those changes and checks for correct timing, typically at each storage or related element encountered in a path. The timing checks are, as noted hereinafter, specified by the end user. Typically, when a storage device such as a flip-flop or latch is encountered, the timing analysis stops a first propagation path and begins a new propagation path at the output of the storage device for reasons to be described hereinafter. In this manner, local or global loops can be easily accommodated. The timing analysis is "aware" of the next simulation time and accordingly, all signal propagation occurring prior to that time is completed, any timing violations are noted, either immediately or saved for later viewing, and thereafter, the next simulation analysis takes place with a timing analysis process occurring thereafter. Thus, the functional simulation and the timing analysis are, in effect, multiplexed by the operation of the present invention. This method of operation is described in greater detail in connection with FIG. 14.

Detailed Description Text - DETX (12):

In accordance with the invention, modules may have their delays specified by the gates and other elements within the module. Nevertheless, at a higher level, a specify block can be employed to override the internal specifications of the module and to assign new, perhaps different, delay characteristics to each of the various paths through the module The specify block can also be employed with an otherwise "empty" module to simply define the delays which are anticipated to be associated with the various paths through the module. In a simple case, the specify block can also be employed to set the signal value relationship between the output and input associated with each path. Such a specify block declaration of a module can be employed as part of an overall design to provide timing continuity between input and output even though the actual functionality of the module has yet to be implemented. Thus, the primary purpose of the specify block is to contain declarations of paths within the block and to provide timing limits with regard to the input and output ports. The specify block also includes timing checks to determine and ensure that the signals maintain a correct timing relationship.

Detailed Description Text - DETX (18):

The path declaration can also employ conditional paths, that is, paths wherein the output specified by the declaration depends upon the signal condition or conditions at one or more of the inputs to the module The condition may be a level sensitive condition, as in the case of a "transparent" latch, or it may be an edge sensitive condition as in the case of a flip-flop. The specify block can handle both types of conditional path in its declaration. In addition, the sensitivity may be different depending upon whether one is describing a positive edge or a negative edge. Accordingly, the declaratory statement can further take this change of directional value into account.

Detailed Description Text - DETX (32):

The delay characteristics of a transparent latch are described in a specify block using two conditional paths, one of which represents a level sensitive path and the other of which represents an edge sensitive path. When the clock is high, the latch is transparent and any change at the input will be propagated to the output. This is the

level sensitive path through the latch and, as noted above, is a conditional path, dependent upon the clock having a high value. In this transparent state, the delay through the latch is straightforward, that is, the delay is specified between the data input and the data output. When the latch "closes," that is, when the clock goes low, a change in value at the input will not be propagated through to the output and instead, the latch waits for the next rising edge of the clock before the data at its input is allowed to propagate through to the output. This is an edge sensitive conditional path through the latch and is specified as a separate path dependent upon the clock rising edge. In this case, it is when the clock changes, not when the data changes, that controls the delay. Thus, a delay path is established between the positive edge of the clock and the output of the latch and the specified delay applies not to the data path but to this clock controlled path.

Detailed Description Text - DETX (33):

A flip-flop is slightly different than the latch, since the flip-flop has no transparent state. The level sensitive path required for the latch is not needed for the flip-flop and accordingly a simple flip-flop can be described with simply an edge sensitive path. However, the flip-flop status becomes more complicated when the clear input to the flip-flop is considered. Thus, for example, to model a D-type flip-flop with an asynchronous clear signal, the specify block must include a conditional path dependent upon the value of the clear. The D-type flip-flop thus contains two edge sensitive paths, and the output of the device can only change in response to a transition at either the clear input or the clock input. Since the clear input takes precedence over the clock input, the edge sensitive path through the clock input must be made conditional upon the state of the clear input. Therefore, an "IF" construction (conditional path) is employed as the first statement in the specify block to limit the effect of a clock pulse to a time when the clear input is not active.

Detailed Description Text - DETX (35):

Among the more important timing checks are the SETUPHOLD check which is used to specify the setup and hold times for storage devices such as latches, flip-flops, and registers. The basic principal behind the setup and hold requirements is that the data signal must remain stable for a specified minimum time period relative to the clock signal transition to successfully store the data (setup) and must further remain stable for a further minimum period in order to hold the data. Any change in the data signal within the specified time limits results in a timing violation. This is illustrated in FIG. 6. Note that the hold time is measured forward in time and the setup time is measured backward in time. This is true when both the setup and hold parameters are positive; however, these parameters can also be expressed as a negative number in order to shift the violation region relative to the clock transition as illustrated in FIG. 7.

Detailed Description Text - DETX (42):

Any of the timing checks previously described can be expressed as a conditional timing check wherein it is activated only upon conditions at other inputs to the module being described. Thus, for example, a latch illustrated in FIG. 12, a D-type latch (flip-flop) 60, is transparent when the clock 62 is high and latches the last value on the data input 64 when the clock drops to low. The preset line 66 forces the output high at any time. The clear line 68 forces the output low at any time. When both are asserted at the same time the clear line overrides the preset line. This storage element can be modeled using two WIDTH timing checks which are dependent upon the clear pulse and the preset pulse, that is, by forcing the clear pulse to have a width greater than a first value and the preset pulse to have a width greater than a smaller value, the clear signal will override the preset and any event at the preset is ignored while the clear is active. Furthermore, since the circuit is not concerned with a preset input under these circumstances, the WIDTH timing check that monitors the preset is made inactive during this time; otherwise, false violations could result. Conditional timing checks are an effective device for reducing the number of false violations produced by the timing analysis. By deactivating a timing check during times when its associated inputs are to be ignored by the storage device, meaningless violations can be avoided and the user is permitted to concentrate on the results which are valid.

Detailed Description Text - DETX (61):

Whenever a path that contributes to a timing violation is identified, the full path, from the primary input to the end point at which the timing check violation occurred, may have been broken several times as a result of traversing various storage devices or a dummy storage module. Each portion of the violation path, between breaks in the path, is designated a path segment and each segment is derived from a separate path trace. For many designs, most paths will consist of two segments, for example, the segment from the clock source to one

flip-flop and the segment from the flip-flop to the data input of another flip-flop. In cases where the clocks are very regular, only the last path segment is important to the user. In some large designs based on transparent latches, paths with large number of segments may exist.

Detailed Description Text - DETX (65):

Most of the time, according to the invention, only one clock cycle will need to be simulated. An exception is when the case analysis, in which the design is simulated for as many cycles as there are cases, is used. Additionally, designs based on transparent latches and designs that employ true multicycle paths will also need to be run for multiple clock cycles.

Detailed Description Text - DETX (74):

A special problem for the timing analysis method results from the use, in a global feedback loop, of solely transparent latches to regulate the critical flow of data around the loop. Thus, while global loops do not present a problem if they contain storage devices that are clock driven, a problem does occur when the devices in the loop are data driven. The normal path analysis will not produce correct results for these data driven loops, primarily because there is no method for being certain where to begin the path tracing method. In accordance with the invention, however, there is provided a special function that can be used in conjunction with the hybrid analysis to guarantee successful analysis of such latch-based designs.

Detailed Description Text - DETX (75):

The latches can thus be used in two different ways, they may be clock driven or data driven depending on how their transparent state is used. As noted above, clock driven latches affect the timing analysis just the way flip-flops do. Their output change is determined by a clock arrival time. Data driven latches are latches which may be in an open (pass through) state when the changed data arrives at the input. As with all storage devices, the system makes the data output of a latch a history point that triggers a path analysis when a change arrives. When that change is traced to the data input of a latch, the system determines if the latch is in the open state. If it is, then the arrival of data at the input causes a history entry at the output just as an opening edge arriving at the clock input will cause such a history entry.

Detailed Description Text - DETX (76):

Thus, a design style, that is sometimes used in designs where the cycle time is critical, permits the critical loops in the design to only have data driven latches. This "style" allows the design to operate with a cycle time that does not have to wait upon the setup time of the latch, and in some of the longer paths, the cycle time can "borrow" time from those paths that precede them. This is a difficult design to verify during the timing analysis. The timing analyzer for such a latch-based design is faced with the chicken and the egg problem. Referring to FIG. 13, in a two phase clock where every path goes from an "A" phase latch to a "B" phase latch, and then back to an "A" phase latch, the analysis of the paths from the "A" phase latches to the "B" phase latch requires the system to know when the data arrives at the "A" phase latch. However, to determine that time, the system must use the path starting at the "B" phase latches. The attempt to find the starting point has thus gone into a loop. The analysis system 14 solves this difficulty by providing means for allowing the user to iterate over the paths until the actual data arrival times have propagated through the design, stabilize, and result in a correct path analysis. Thus, referring to FIG. 13, the analysis system 14 only has to loop twice through the circuitry of FIG. 13 if the timing is correct. After the first time, the system evaluates the "B" phase latch, and the timing of the "A" phase latch will then be correct. However, it will have to loop a third time to ensure that two passes through the loop was sufficient. It can determine this by noting that no arrival times have changed at the latches between the second and third loops.

Detailed Description Text - DETX (77):

If the timing is still incorrect, the analysis can continuously loop many times, with the arrival time of the data advancing later in each cycle, until the data arrives after one of the latches is in the latched state. This will provoke a timing violation, and the loops will stabilize since the latch will not have changed any later then when the clock input changed to the latched state.

Detailed Description Text - DETX (78):

The system thus provides a latch change function to enable the user to set up the proper iterations for the analysis of a latch-based design. This function returns a value that only changes when a latch experiences a change on its data input during an open state and only if that change arrives later than it did during the previous cycle. The user should have the system loop, simulating cycles of the design, until no latches cause a "latch change" to change. In particular, the latch change function returns the difference in time from the maximum arrival of an opening clock edge to the maximum arrival of a data change, for any latch in the design, for which the interval has become larger and for which the difference is smaller than the value that the "latch change" would otherwise return. The user initializes the return value of the "latch change" with an argument provided to it. By initializing the command to a large value each cycle, the system will determine and identify if any latches saw a later data arrival time than on the previous cycle.

Claims Text - CLTX (16):

providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing at least one transparent latch.

Claims Text - CLTX (29):

providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing at least one transparent latch.

Claims Text - CLTX (41):

providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing at least one transparent latch.

Claims Text - CLTX (50):

providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing at least one transparent latch.

Claims Text - CLTX (55):

providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing at least one transparent latch, and

Claims Text - CLTX (68):

providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing at least one transparent latch, and

Claims Text - CLTX (85):

means for providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing transparent latches.

Claims Text - CLTX (99):

means for providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing transparent latches.

Claims Text - CLTX (111):

means for providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing transparent latches.

Claims Text - CLTX (120):

means for providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing transparent latches.

Claims Text - CLTX (125):

means for providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing transparent latches, and

Claims Text - CLTX (138):

means for providing for iteration of stimuli at designated inputs for checking the timing of a digital circuit design having a loop containing transparent latches, and